



CENTRE DE ROCQUENCOURT

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
B.P.105  
78153 Le Chesnay Cedex  
France  
Tél: 954 90 20

350 se. le 7. 12. 81.

# Rapports Techniques

N° 6

## COUPLAGE MITRA - INTERDATA

Jean-Paul CHIÈZE  
François EYRIÈS  
Serge LÉVY

Novembre 1981

COUPLAGE MITRA-INTERDATA

Jean-Paul Chièze (INRIA)

François Eyriès (INRIA)

Serge Lévy (IITN)

*5 me quai de la Seine Eiffel.*

*91420 HORANGIS.*

Résumé:

Nous présentons un rapport d'ensemble sur le couplage réalisé entre l'INTERDATA 8/32 et le MITRA 125 du groupe graphique INRIA.

Abstract:

This report presents the way we used to connect two INRIA systems: INTERDATA 8/32 and MITRA 125.

INDEX

1-Le contexte.....	3
2-Buts .....	4
3-Solution.....	5
4-Protocole.....	6
5-Réalisation.....	12
6-Exemples.....	13

=====

1-LE CONTEXTE

Le groupe graphique dispose d'un MITRA 125 sous système SALUT à usage général, c'est-à-dire que nous développons nos textes avec l'éditeur de texte, compilons via fortran ou pascal, éditons les liens et exécutons sur le MITRA même.

La configuration est la suivante:

- 1 UC 125 avec flottant microprogrammé;
- 256 KO de mémoire;
- 1 tourne-disque 50 MO sur IOP (processeur d'entrée/sortie) ;
- 1 tourne-disque 2\*5 MO sur IOP;
- 1 dérouleur de bande 1600 BPI sur UC;
- 1 tourne-disquette sur UC;
- 1 lecteur de cartes sur UC;
- 1 imprimante point par point VERSATEK 2200 points/ligne sur AMC (coupleur 16 bits parallèle) ;
- 1 console graphique au trait TEKTRONIX 4014 sur AMC;
- 1 console vidéo couleur TEKTRONIX 4027 sur ligne asynchrone;
- 5 terminaux alphanumérique sur lignes asynchrones.

Ce MITRA devient insuffisant sur les points suivants:

- espace d'adressage des programmes générés par fortran ou pascal limité à 64 KO, d'où beaucoup de temps perdu en segmentation;
- amplitude des entiers limités à 16 bits;
- précision des flottants limités à 24 bits de mantisse;
- lenteur de l'UC;
- lenteur du code généré par les deux compilateurs fortran et pascal.

L'INTERDATA 8/32 sous CS32 dispose de:

- 360 KO de mémoire directement adressable par 1 seul programme;
- 1 UC 8/32 avec instructions sur entiers 32 bits et flottants 32 et 64 bits;
- 1 tourne-disque 2\*5 MO sur SELCH (processeur d'entrée/sortie) ;
- un optimiseur dans les compilateurs fortran et pascal.

Des tests à partir de sources fortran montrent un rapport de performance de 12. Dès lors, l'exploitation suivante s'impose:

- édition des textes, programmes et données sur le MITRA;
- compilation et exécution sur l'INTERDATA, sachant que la quasi totalité des supports (périphériques et fichiers) seront sur MITRA.

Pour connecter les 2 machines:

- 1 AMC (16 bits parallèle avec mode canal) sur MITRA;
- 1 SELCH et 1 ULI (16 bits parallèle) sur INTERDATA.

2-BUTS

- Accéder depuis INTERDATA à n'importe quel support (périphérique ou fichier) MIRA;
- Même programme source qu'il soit à compiler sur MIRA ou sur INTERDATA, pour ce qui concerne l'accès aux supports;
- Utiliser les tâches (programmes exécutables) INTERDATA existantes, ce qui entraîne d'accéder aux supports MIRA via un ordre d'IO standard (SVC1 de l'OS32) et non par routines spéciales;
- Parallélisme des accès par une ou plusieurs tâches INTERDATA;

- EX1: compiler sur INTERDATA un fichier source se trouvant sur disque MIRA;
- EX2: lecture par l'IM (la tâche temps partagé de l'OS32) de commandes sur terminal ou fichiers MIRA;
- EX3: sortir les listings spoolés sur VS (l'imprimante MIRA) ;
- EX4: exécuter une tâche INTERDATA avec interaction sur un terminal MIRA;
- EX5: analyser ou positionner une bande sur dérouleur MIRA via les commandes OS32 ou MIM;
- EX6: back-up des disques 5 MC INTERDATA sur bande MIRA et restauration;
- EX7: exécution parallèle d'une tâche INTERDATA et d'une tâche MIRA (ou INTERDATA) communiquant par IOS standards;
- EX8: changement de support MIRA en cours d'exécution d'une tâche INTERDATA.

En conséquence, un couplage au niveau fichier, c'est-à-dire transfert aller avant d'exécuter une tâche INTERDATA, transfert retour après, est insuffisant: impossible de traiter EX2, EX4, EX5, EX7, EX8 par exemple.

Nous avons donc spécifié un couplage au niveau bloc d'IO: lire un bloc, écrire un bloc, se positionner dans un fichier.

=====

3-SOLUTION

20 voies logiques sont multiplexées entre un driver INTERDATA d'un côté, une tâche MITRAX de l'autre.

Du côté INTERDATA, nous déclarons à la génération du système CS32, 20 devices V0 à V19, attachés au driver INTERDATA.

Du côté MITRA, la tâche MITRAX maintient en permanence 19 unités logiques U:1 à U:19.

31-Comment exécuter un IC distant?

Un ordre d'IC sur l'un des devices V0 à V19 arrive à INTERDATA sous forme d'un bloc d'IC standard CS32. INTERDATA recopie ce bloc dans une demande qu'il envoie à MITRAX. MITRAX interprète la demande et renvoie la réponse. INTERDATA recopie la réponse dans le bloc d'IC et rend IC terminé à CS32.

L'ensemble INTERDATA-MITRAX se comporte donc comme un driver à distance gérant un ensemble de supports. Les 20 voies logiques sont multiplexées sur la liaison INTERDATA-MITRA de façon qu'aucune ne bloque les autres.

32-Comment assigner un support MITRA depuis INTERDATA?

Pour assigner un support distant depuis INTERDATA, il faut faire en un bloc les 2 assignations suivantes:

- assigner un device parmi les devices libres de V1 à V19 avec la commande standard ASSIGN de MIT;
- assigner l'unité logique de même indice parmi U:1 à U:19 dans MITRAX.

a-assigner au chargement

On dispose de la macrocommande:

MASSIGN <unité logique>, <support MITRA>

Cette macro:

\*assigne le premier device libre de V10 à V19; soit Vi ce device;

\*envoie sur V0 une chaîne alphanumérique de la forme:

ASSIGN,U:i,I:<support MITRA>

Le device V0 et la voie 0 sont réservées. Il n'y a pas d'unité logique U:0. MITRAX exécute directement, par appel au moniteur SALUT, les commandes qui arrivent sur la voie 0. Si SALUT renvoie assignation bonne, MITRAX renvoie ok, sinon ordre illégal.

En résumé, assigner un support distant revient à remplacer une commande ASSIGN par une commande MASSIGN dans les jobs.

EX: MASSIGN 3,"VS,AN"

\*cherche un device libre, par exemple, V10;

\*envoi sur VO l'assignation MITRA:

ASSIGN,U:10,T:VS,AN

b-assigner à l'intérieur d'un programme dont on possède la source.

On dispose de l'integer function MASSIGN (<unité logique>, <support MITRA>) qui exécute le même algorithme que la commande MASSIGN et rend 0 si ok, 1 sinon.

c-assignation à l'intérieur d'un programme dont on ne possède pas la source.

On dispose aussi de la commande:

MASSIGN <V1 à V9>,<assignation MITRA>

qui effectue seulement l'assignation côté MITRA. Dans ce cas, l'exécution en un bloc des 2 assignations n'est pas assurée.

Note: toute écriture sur le device VO est considérée par MITRAX comme une commande au moniteur SALUT. Par ce moyen, on peut envoyer un job à MITRA depuis INTERDATA. Pour chaque écriture sur VO, s'il y a erreur de commande, MITRAX renvoie ordre illégal.

=====

#### 4-PROTOCOLE

##### 41-Niveau appareil

###### 411-Demande et réponse

Format de la demande.

La demande est envoyée en 1 transfert ou 2 transferts successifs:

-partie fixe de 16 octets contenant

2 octets: IENVOI (cf. Niveau transport)

2 octets: IATTENDU (cf. Niveau transport)

1 octet : numéro de station émission (cf. Niveau transport)

1 octet : numéro de station réception (cf. Niveau transport)

1 octet : numéro de voie logique

1 octet : function code

4 octets: random address

4 octets: length of data

-partie variable de length of data octets si le function code est un WRITE ou un TRANSPARENT (cf ci-dessous)

Format de la réponse

La réponse est renvoyée en 1 transfert ou 2 transferts successifs:

-partie fixe de 12 octets contenant:

2 octets: MENVOI

2 octets: MATTENDU

1 octet : numéro de station émission

1 octet : numéro de station réception

1 octet : device dependant status

1 octet : device independant status

4 octets: real length

-partie variable de longueur real length si l'ordre est un READ ou un TRANSPARENT.

Signification:

\*function code, random address, length of data sont recopiés tels quels par INTERDATA depuis le bloc d'IO INTERDATA;

\*device dependant status, device independant status, real length sont à recopier tels quels par INTERDATA dans le bloc d'IO INTERDATA;

\*c'est-à-dire que l'interprétation de l'IO est faite dans MJTRAX;

##### 412-Les ordres



## Couplage MITRA-INTERDATA

MITRAX traite les 18 ordres suivants (avec les fonction codes correspondants en hexadécimal) :

WRITE alphanumérique séquentiel formaté	20
WRITE alphanumérique séquentiel image	21
WRITE alphanumérique direct formaté	24
WRITE alphanumérique direct image	25
WRITE binaire séquentiel	30 ou 31
WRITE binaire direct	34 ou 35
READ alphanumérique séquentiel formaté	40
READ alphanumérique séquentiel image	41
READ alphanumérique direct formaté	44
READ alphanumérique direct image	45
READ binaire séquentiel	50 ou 51
READ binaire direct	54 ou 55
REWIND	C0
BACKSPACE RECORD	A0
FORWARDSPACE RECORD	90
WRITE FILE MARK	88
FORWARD SPACE FILEMARK	84
BACKSPACE FILEMARK	82

### Signification:

- \*alphanumérique: les data sont transcodées ASCII-EBCDIC dans MITRAX et l'assignation MITPA doit contenir le champ AN. Ce champ précise au handler MITRA qu'il doit transcoder ou non depuis EBCDIC suivant le support;
- \*binaire: pas de transcodage et l'assignation MITRA doit contenir le champ BN;
- \*séquentiel: l'accès concerne le bloc suivant du bloc actuel sur le support;
- \*direct: le numéro de secteur est contenu dans le champ RANDCN de la demande;
- \*formaté: la lecture ou l'écriture s'arrête au premier RC (non compris) , avec une longueur maxima en octets donnée dans le champ length of data de la demande;
- \*image: la lecture ou l'écriture a une longueur en octets égale à length of data de la demande;

Ces ordres sont ceux générables, par exemple, via un compilateur fortran ou pascal. La plupart des tâches standard fournies avec l'OS32 n'utilisent que ce lot d'ordres et fonctionnent correctement avec un support MITPA.

Un 19ème ordre est reconnu:

TRANSPARENT

C1

## Couplage MITRA-INTERDATA

(la fonction code 81 est réservé sous OS32 aux driver dependent functions)

Pour cet ordre, MITRAX exécute un bloc d'IO format MITRA, fourni en début de data. Cet ordre permet d'accéder aux fonctions des handlers MITRA non traitées par les 18 autres.

L'ensemble des codes-retour MITRA est partitionné dans les classes suivantes:

00	ok
A0	support pas prêt
90	fin de volume
C0	ordre illégal
84	erreur irrécupérable
82	erreur récupérable
88	fin de fichier
98	fin de fichier+fin de volume

Lorsque l'ordre n'a pas de sens avec l'assignation courante, MITRAX renvoie une réponse contenant ordre illégal (C0) .

### Ordres traités vs les supports MITRA.

/Supports fichier zoneDA bande versatek tektro4014 terminaux cartes boitespost.

noms :	Dx	Cx	9T	VS	TK	Tx	CR	BP
Ordres								
WR,A,S,F	1		1	1	1	1		1
WR,A,S,I	1		1	1	1	1		1
WR,A,D,F		1						
WR,A,D,I		1						
WR,B,S	1		1	1	1			1
WR,B,D		1						
RE,A,S,F	1		1		1	1	1	1
RE,A,S,I	1		1		1	1	1	1
RE,A,D,F		1						
RE,A,D,I		1						
RE,B,S	1		1		1		1	1
RE,B,D		1						
REWIND	1		1					
FFILE			1					
BFILE			1					
FRECORD			1					
BRECORD			1					
FILEMARK	1		1					

Aucune vérification n'est faite pour l'ordre TRANSPARENT (à utiliser avec précaution) .

#### 42-Niveau transport

IENVOI est le numéro d'ordre de la demande incrémenté de 1 à chaque demande.

IATTENDU est le numéro de la réponse que veut recevoir INTERDATA, soit la plus ancienne réponse non reçue par INTERDATA.

MENVOI est le numéro d'ordre de la réponse incrémenté de 1 à chaque nouvelle réponse.

MATTENDU est le numéro de la demande que veut recevoir MITRAX, soit la plus ancienne demande non reçue par MITRAX.

Par ce mécanisme, une demande (une réponse) perdue est automatiquement redemandée en priorité. La perte ne peut résulter que d'une erreur de transfert au niveau routage.

Les champs numéro de station émission, numéro de station réception sont réservés.

Les 20 voies fonctionnent indépendamment.

Le séquençement sur une voie est le fait de l'OS32 qui n'autorise qu'1 IO courant par device. Pour la même raison, il n'y a pas d'anticipation sur une voie: la tâche INTERDATA à intérêt à travailler de façon asynchrone.

#### 43-Niveau routage

MITRA et INTERDATA sont reliés via 2 coupleurs type parallèle 16 bits:

-AMC sur MITRA (débit maximum si aucun autre AMC ne travaille: 200 KO/sec) ;

-SELCH & ULI sur INTERDATA (débit maximum: 1 MO/sec) ;

Ces 2 coupleurs communiquent par un câble et un interface matériel installé sur la carte ULI. L'interface traduit la logique périphérique AMC en logique périphérique ULI et inversement: 32 signaux de données (16 aller, 16 retour), 11 de commande.

Nous utilisons le mode canal sur les 2 coupleurs.

a-Emission INTERDATA->MITRA

INTERDATA envoie une II (interruption) au MITRA qui est mémorisé par une séquence d'interruption attachée à l'AMC.

MITRAX lance alors le canal de l'AMC en lecture.

b-Emission MITRA->INTERDATA.

MITRAX lance son canal en écriture.

Dans les 2 cas c'est MITRAX qui lance le premier son canal. Le départ du canal de l'AMC produit une IT qui active INTERDATA, qui lance alors son canal.

Un conflit intervient si INTERDATA génère une IT alors que MITRAX est déjà dans sa séquence d'émission. C'est pourquoi INTERDATA teste, lorsqu'il reçoit l'IT, si le canal AMC est parti en lecture ou en écriture.

=====

## 5-REALISATION

### 51-Actuel

La réalisation a demandé au total environ 1 an\*ingénieur:

- spécifications;
- réalisation de l'interface matériel inter-coupleurs;
- programmation de MITRAX et INTERDATAx;
- recettes et intégrations.

Quelques caractéristiques.

- MITRAX: 1000lignes fortran IV, 200 lignes assembleur, 10 KO non compris les buffers;
- INTERDATAx: 200 lignes assembleur, 1 KO.

Contributions:

- J.P.Chièze: conception et programmation d'INTERDATAx;
- F.Eyriès: spécifications ; conception et programmation de MITRAX;
- S.Lévy: conception et câblage de l'interface matériel.

### 52-Additions réalisables

- a-Connecter MITRA à un autre système, concouramment à INTERDATA:
  - additions dans MITRAX;
  - écrire un driver distant analogue à INTERDATAx.
- b-Porter MITRAX sur un autre système, sur MITRA ou autre.
- c-routage multisites.

=====

6-EXEMPLES

Rappelons que CS32 dispose d'une tâche temps-partagé MTM capable d'exécuter en parallèle plusieurs flois de commandes en mode interactif ou en mode batch. D'autre part, on peut regrouper les commandes en macrocommandes, dites fichiers CSS, avec substitution de paramètres.

EX1-Comment exploiter MTM en interactif depuis un terminal MITRA, dans l'exemple IO?

Au pupitre CS32, faire:

MASSIGN V1:,"IO,AN"

.MTM ADD V1:

Note: on peut regrouper en une seule commande CSS:

MTMADD IO

EX2-Comment exploiter MTM en batch depuis un fichier MITRA, dans l'exemple le fichier MTM-COMMANDS.

Au pupitre CS32, faire:

MASSIGN V2:,"MTM-COMMANDS,AN"

.MTM SUBMIT V2:

Notes: -la tâche chargée par le job MTM-COMMANDS peut être interactive.

-on peut regrouper en une macro:

MTMSUBMIT MTM-COMMANDS

EX3-Compiler, linker, exécuter sur INTERDATA un source fortran sur disque MITRA.

Le fichier disque MITRA : GESTION-VERIFIER contient un programme fortran qui lit un article sur l'étiquette logique 1, lit un article sur l'étiquette 2, écrit sur l'étiquette 3 si les 2 articles diffèrent, et ainsi de suite jusqu'à la première fin de fichier. On veut compiler avec le listing sur l'imprimante MITRA VS. On veut exécuter avec les assignations:

1=le fichier disque MITRA GESTION-RECORDS

2=la bande magnétique MITRA 9T

3=l'imprimante MITRA VS

Solution a-En interactif depuis IO, envoyer à MTM:

LOAD FORTRAN

\*fortran lit sur 1, liste sur 3, produit sur 2

MASSIGN 1,"GESTION-COMPARER,AN"

ASSIGN 2,VERIFIER.OBJ

MASSIGN 3,"VS,AN"

```

START
LOAD TET32
*édition de liens
START
ESTAB TASK
GET 500
INCLUDE VERIFIER.OBJ
EDIT F7RTL.OBJ/S
BUILD TASK,VERIFIER.TSK
END
LOAD VERIFIER.TSK
MASSIGN 1,"GESTION-RECORDS,AN"
MASSIGN 2,"9T,AN"
MASSIGN 3,"VS,AN"
START

```

#### Solution b-En batch

Plutôt que réécrire plusieurs fois le même job à MTM, mieux vaut le mémoriser dans un fichier :

- soit sur MITRA et l'envoyer à MTM comme vu dans EX2;
- soit sur INTERDATA et l'exécuter :
  - soit en mode CSS;
  - soit en mode batch par SUBMIT;

#### EX4-Travail en parallèle d'une tâche INTERDATA et d'une tâche MITRA

La tâche INTERDATA PICTURE.TSK lit des données sur l'étiquette 1, des paramètres sur l'étiquette 2 et produit des pixels sur l'étiquette 3. La tâche MITRA TVCOLOR lit ces pixels en pipeline sur l'étiquette SI et affiche une image sur l'étiquette LC.

Avec :

- 1=le fichier disque MITRA 3D-MAQUETTE
- 2=le terminal d'où vient le job pour MTM
- 3 et SI=la boîte postale BP
- LC=le tektronix couleur 4027 TV

a-Mettre au batch MITRA le job :

```

JOB TVCOLOR
NIVEAU,N:2
ASSIGN,M:SI,I:BP
ASSIGN,M:LC,I:TV
CHARGER TVCOLOR

```

b-Lorsque TVCOLOR est chargé, envoyer à MTM :

```
LOAD PICTURE.TSK
MASSIGN 1,"3D-MAQUETTE,BN"
MASSIGN 3,"BP,AN"
MASSIGN 2,CON:
START
```

D'autres exploitations existent pour arriver au même résultat.

EX5-Accès direct à un fichier MITRA.

La tâche INTERDATA HERSHEY lit sur l'étiquette 1 une chaîne alphanumérique qu'elle écrit sur l'étiquette 2 en utilisant une police de caractères se trouvant sur l'étiquette 3, accordée en direct.

Avec: 1=le tektronix 4014 TK en alphanumérique  
 2=le tektronix 4014 TK en binaire  
 3=la zone DA du disque DC

Envoyer à MTM:

```
LOAD HERSHEY
MASSIGN 1,"TK,AN"
MASSIGN 2,"TK,BN"
MASSIGN 3,"DC,BN"
START
```

EX6-Compiler sur INTERDATA le 3ème fichier d'une bande MITRA

Monter la bande sur 9T et envoyer à MTM:

```
LOAD FORTRAN
MASSIGN 1,"9T,AN"
REWIND 1
FFILE 1
FFILE 1
etc
```

=====



Imprimé en France  
par  
l'Institut National de Recherche en Informatique et en Automatique